XComment.io - Comments Remover Plugin

This plugin removes comments from source files for a number of programming languages. Required Python 3.6 on Jenkins server. A new build step is added: 'Invoke Comments Remover' which accepts files to process as input and creates uncommented version of them.

User can specify Python path in global settings for the plugin (otherwise the one on the system PATH is used).

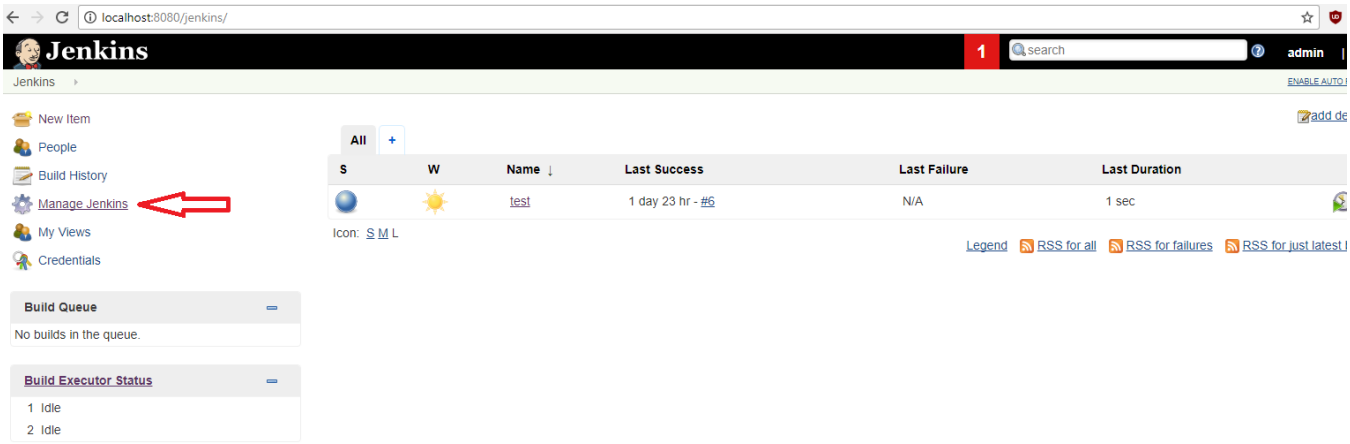
Contributors

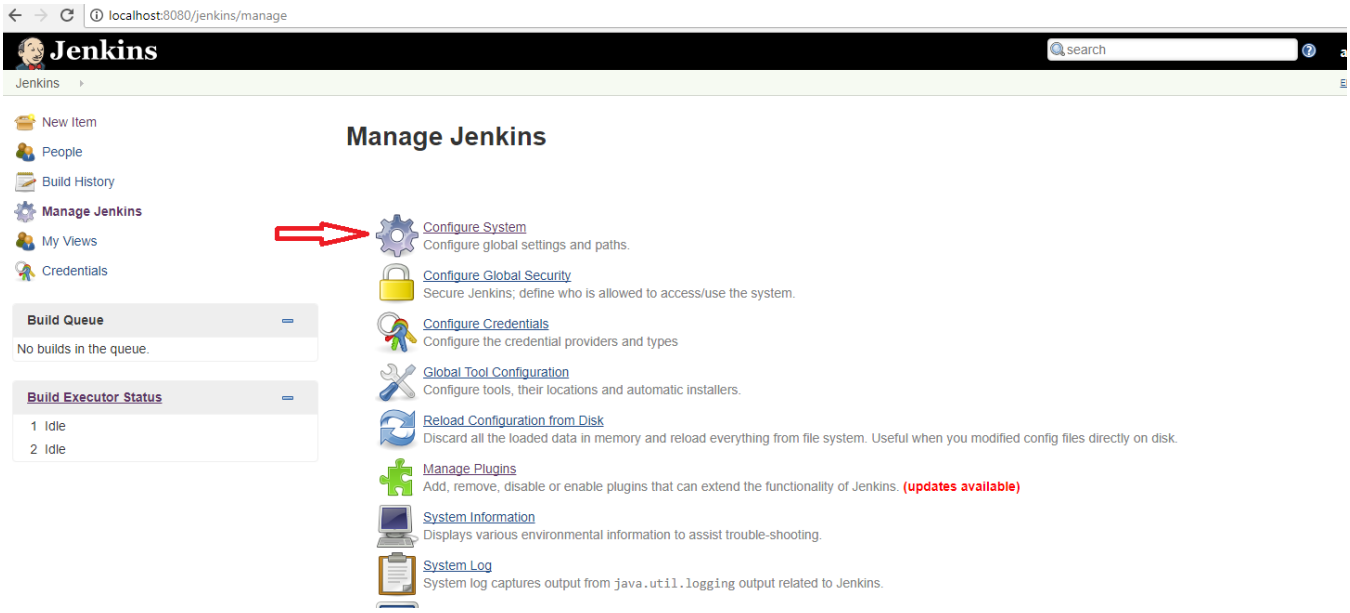
Daniel Dylg

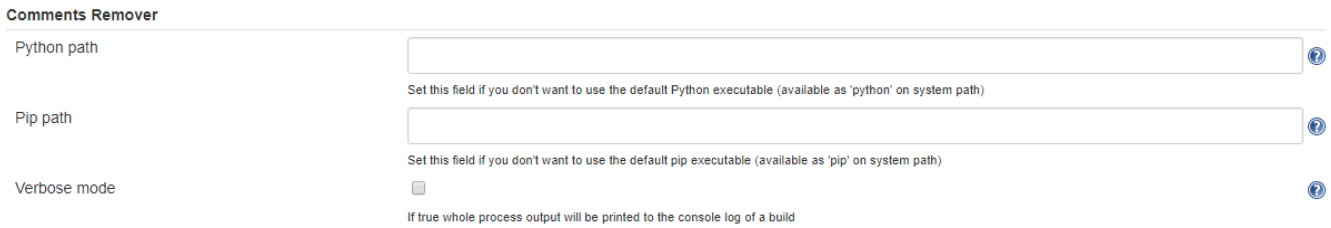
For users

Global settings

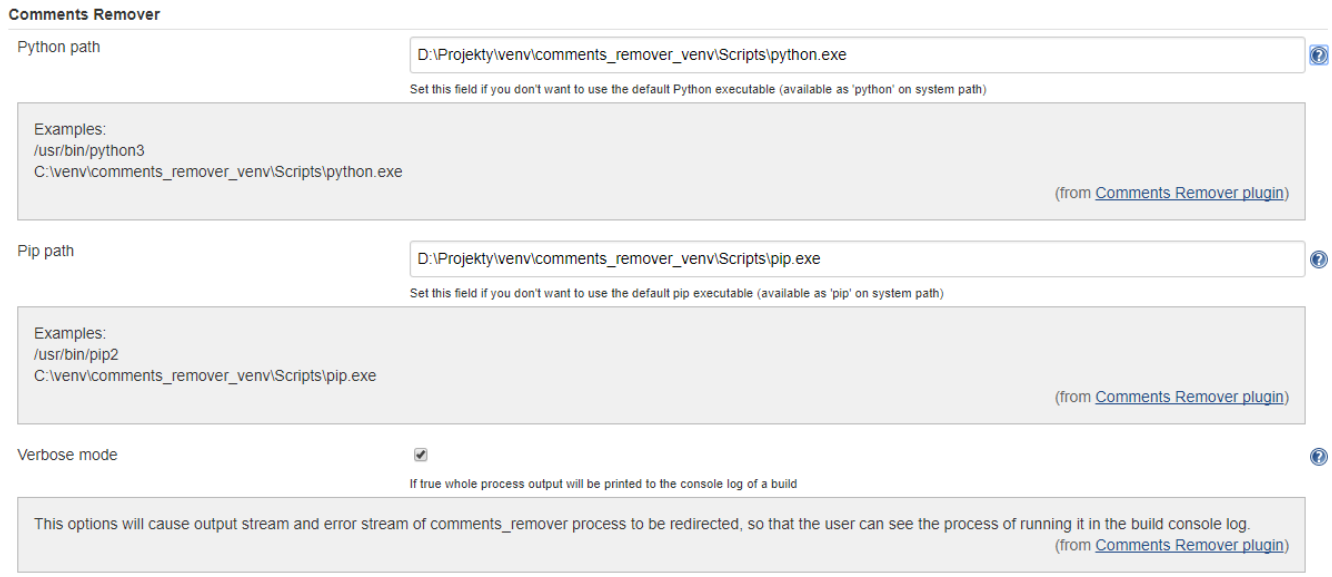
Go to Manage Jenkins -> Configure System to access them





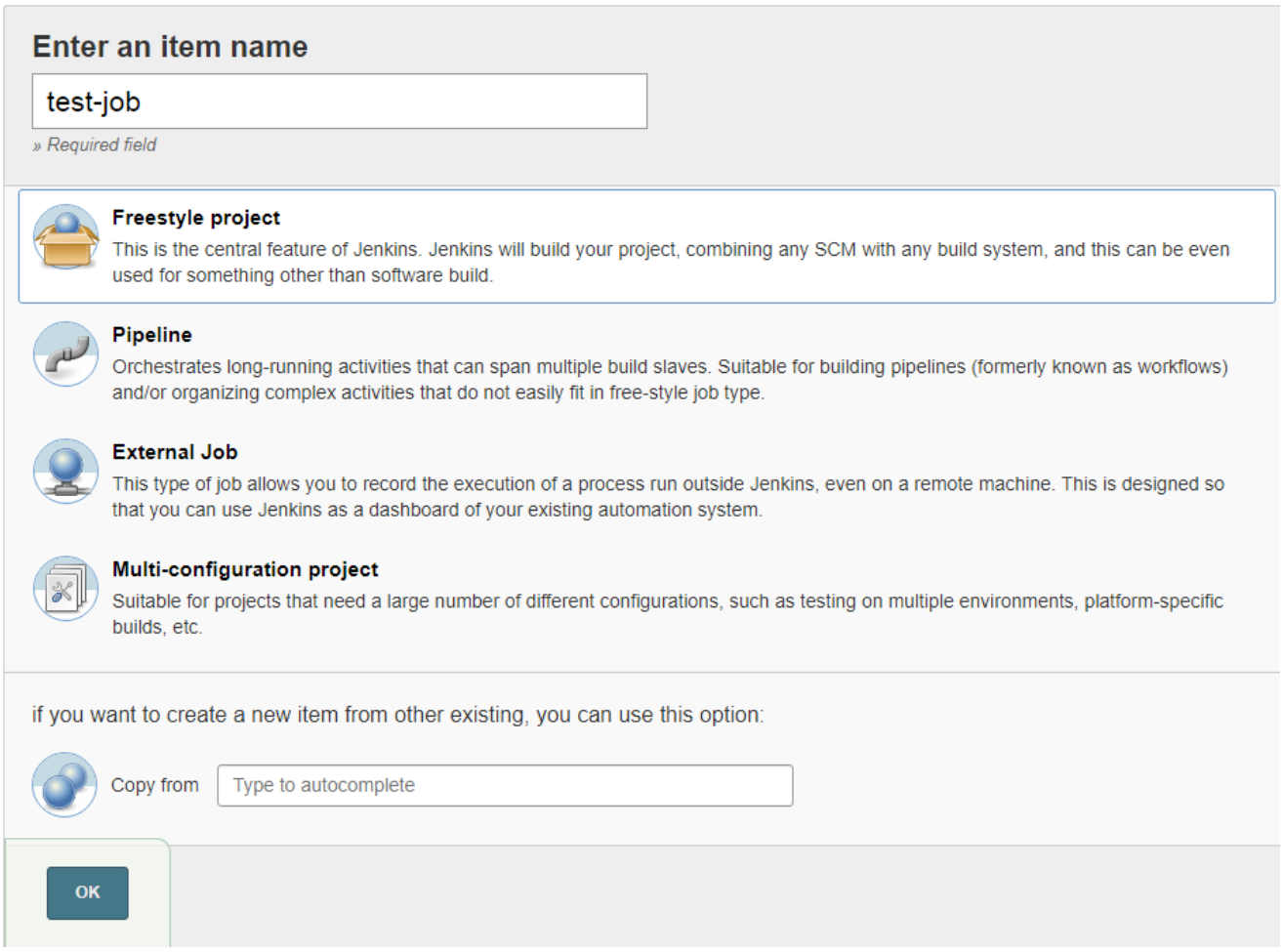


There is help section to provide examples:

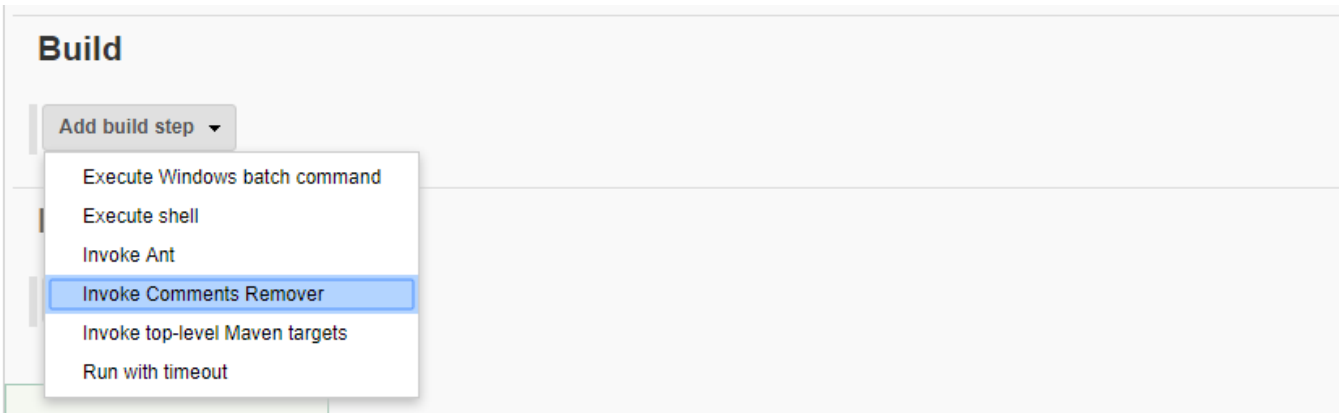


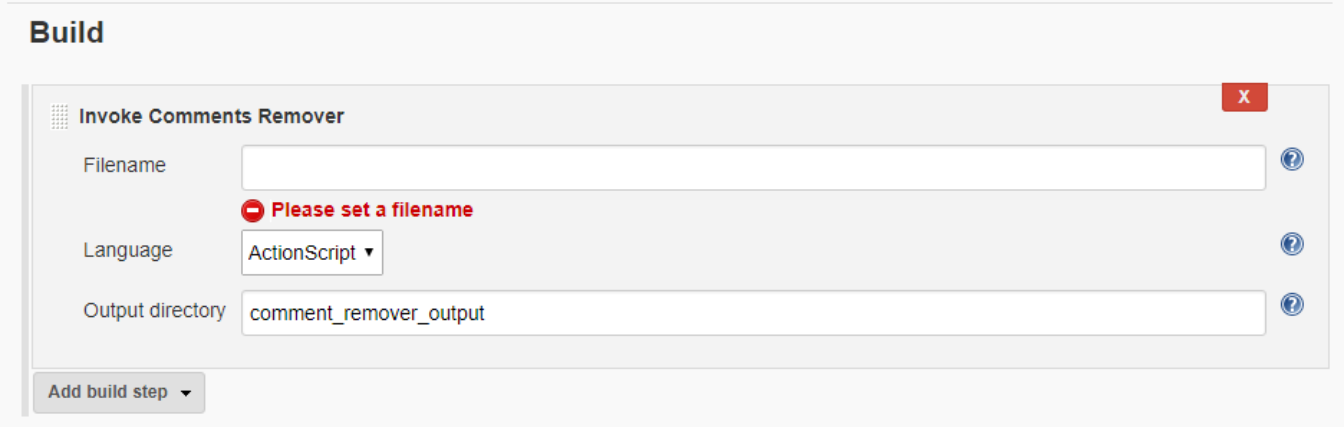
Usage

The plugin can be used wherever build steps can be defined, e.g. Freestyle project

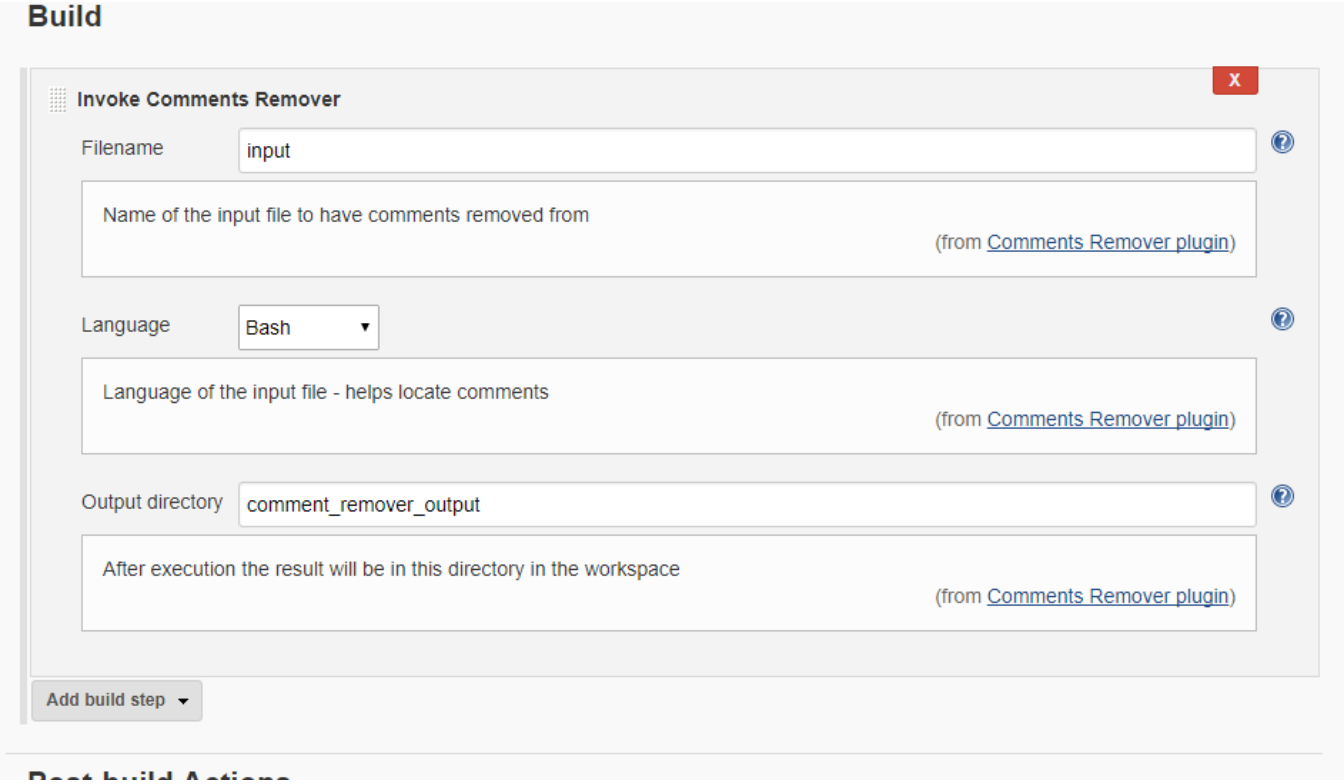


Adding build step:

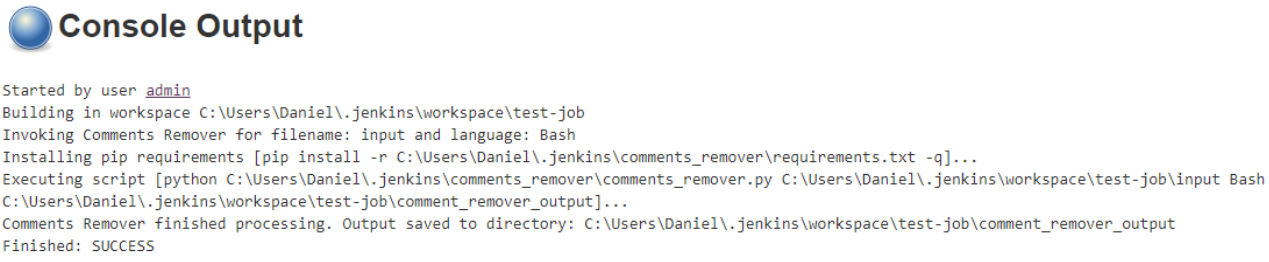


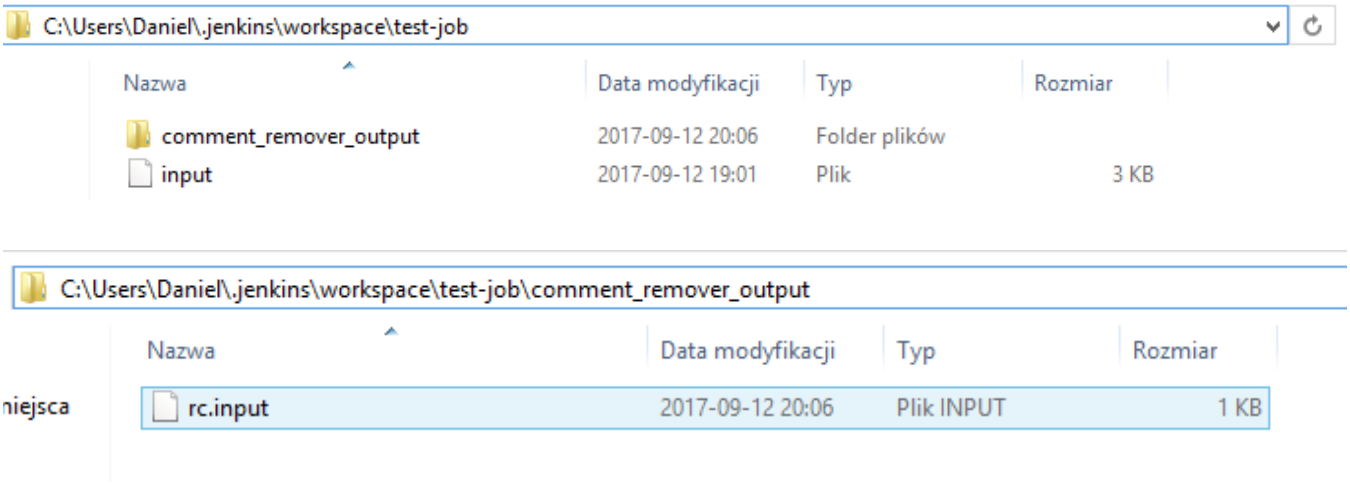


This part also has "help" section:



Example of build output and result:





Using as a shell/batch command script

Alternatively, user can create plain script build step (`Execute shell` or `Execute Windows batch command`) and run `comments_remover.py` Pytho n script manually.

Comments Remover script is unpacked to JENKINS_HOME directory (which is available in script as environmental variable)

If you use Jenkins Struct Plugin, this plugin has a shorthand defined: `commentsremover`.

For contributors

Updating plugin with a new version of Comments Remover

Create ZIP archive of all files required to run Comments Remover - in particular, `comments_remover.py` and `requirements.txt`. Put it in `src/main/resources`. The plugin expects to find `comment_remover.py` and `requirement.txt` on the top lever of the archive.

Debug

```
mvnDebug hpi:run
```

You can attach Java Debugger to a local Java process of Jenkins.

Prepare plugin for distribution

```
mvn package
```

The *.hpi file will be created in target/ directory. To install manually on local Jenkins, copy it to $JENKINS_HOME/plugins directory.

Releasing new version to Jenkins Update Center

```
mvn release:prepare release:perform
```

Follow the instructions to assign new version number.

You must have an account on jenkins-ci.org with commit rights, and its credentials configured in maven settings:

```
<server>
  <id>maven.jenkins-ci.org</id>
  <username>...</username>
  <password>...</password>
</server>
```

More info here:

https://wiki.jenkins.io/display/JENKINS/Hosting+Plugins

For contributors

Updating plugin with a new version of Comments Remover

Create ZIP archive of all files required to run Comments Remover - in particular, `comments_remover.py` and `requirements.txt`. Put it in `src/main/resources`. The plugin expects to find `comment_remover.py` and `requirement.txt` on the top lever of the archive.

Debug

```
mvnDebug hpi:run
```

You can attach Java Debugger to a local Java process of Jenkins.

Prepare plugin for distribution

```
mvn package
```

The *.hpi file will be created in target/ directory. To install manually on local Jenkins, copy it to $JENKINS_HOME/plugins directory.

Releasing new version to Jenkins Update Center

```
mvn release:prepare release:perform
```

Follow the instructions to assign new version number.

You must have an account on jenkins-ci.org with commit rights, and its credentials configured in maven settings:

```
<server>
  <id>maven.jenkins-ci.org</id>
  <username>...</username>
  <password>...</password>
</server>
```

More info here:

https://wiki.jenkins.io/display/JENKINS/Hosting+Plugins

For contributors

Updating plugin with a new version of Comments Remover

Create ZIP archive of all files required to run Comments Remover - in particular, `comments_remover.py` and `requirements.txt`. Put it in `src/main/resources`. The plugin expects to find `comment_remover.py` and `requirement.txt` on the top lever of the archive.

Debug

```
mvnDebug hpi:run
```

You can attach Java Debugger to a local Java process of Jenkins.

Prepare plugin for distribution

```
mvn package
```

The *.hpi file will be created in target/ directory. To install manually on local Jenkins, copy it to $JENKINS_HOME/plugins directory.

Releasing new version to Jenkins Update Center

```
mvn release:prepare release:perform
```

Follow the instructions to assign new version number.

You must have an account on jenkins-ci.org with commit rights, and its credentials configured in maven settings:

```
<server>
  <id>maven.jenkins-ci.org</id>
  <username>...</username>
  <password>...</password>
</server>
```

More info here:

https://wiki.jenkins.io/display/JENKINS/Hosting+Plugins